



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

09/881,791

06/14/2001

Arzhan I Kinzhalin

SUNMP016

1614

25920

7590

07/27/2006

MARTINE PENILLA & GENCARELLA, LLP
710 LAKEWAY DRIVE
SUITE 200
SUNNYVALE, CA 94085

EXAMINER

STEELMAN, MARY J

ART UNIT

PAPER NUMBER

2191

DATE MAILED: 07/27/2006

Please find below and/or attached an Office communication concerning this application or proceeding.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

MAILED

JUL 27 2006

Technology Center 2100

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 09/881,791
Filing Date: June 14, 2001
Appellant(s): KINZHALIN ET AL.

Kenneth D. Wright, Reg. No. 53,795
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 18 April 2006 appealing from the Office action
mailed 16 June 2005.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6,332,211 B1

PAVELA

12-2001

Microsoft Press Computer Dictionary Third Edition, p. 28, 1997.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 1-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Pavela (US 6,332,211, hereinafter Pavela) in view of Microsoft Press Computer Dictionary (MPCD)

With respect to claim 1, Pavela discloses a method for automated acquisition of assertions in a specification of a computer program:

receiving an input specification (source file), wherein the input specification comprises a plurality of sentences (commands or tags) (E.g., see FIG. 2:202, 01. 2:12-17);

obtaining a sentence (tag or a test code object) from the plurality of sentences; (E.g., see FIG. 5:504, col. 6:4);

determining whether the obtained sentence is a testable (executable test code object) assertion (tag), wherein the testable assertion describes behavior of program that can be tested (E.g., see col. 6:28-30, tags are associated with the members of the library of executable test code object; and

marking (indexing) the obtained sentence as testable when the obtained sentence is a testable assertion, using the sentences marked as testable to determine whether a test suite for

testing the computer program is adequate (E.g., see FIG. 3:320, col. 6:47 a test index is automatically generated from the identified system elements).

Pavela does not disclose application programming interface.

However, MPCD discloses application programming interface (p. 28). Therefore, it would have been obvious to incorporate the teaching of MPCD into Pavela to have the testable assertion describes behavior of application programming interface that can be tested because one would want to test application programming interface as well as other sections of the computer program.

As per claim 2, Pavela also discloses identifying a context (tags) within the specification (E.g., see FIG. 7:702, col. 7:23-44).

With respect to claim 3, Pavela further discloses a method wherein the operation of obtaining the sentence from the plurality of sentences includes parsing (scanning and identifying) the context to obtain the sentence (E.g., see FIG. 7:706, col. 14:24-25, performing the step of scanning the interpreted tags to identify the system elements).

With respect to claim 4, Pavela also discloses a method further comprising the operation of adding (incorporating) the marked obtained sentence to an assertion result set (E.g., see col. 11:30-32, a test index identifying system elements test code is generated and incorporated into the test plan).

With respect to claim 5, Pavela further discloses a method wherein the context is a set of circumstances (configuration) related to the obtained sentence (E.g., see col. 5:48-50, test procedures, verification, system configuration, and the parts used by the test case are entered).

With respect to claim 6, Pavela further discloses a method wherein each assertion includes at least one sentence of the specification (E.g., see col. 5:5-7, each tag is associated with a member of a library of executable code objects).

With respect to claim 7, Pavela further discloses a method wherein each assertion can comprises at least two sentences of the specification (See FIG. 52502 and 5:504, wherein test scenario is specified by the plurality of lines; col. 2:3245, 11 :24-26, each tag is associated with a member of a Library of executable code objects defining a set of instructions).

With respect to claim 8, Pavela discloses automatically obtaining assertions from a specification for a computer program, comprising: a code segment that receives the specification as an input (E.g., see col. 2:12-17, this claim is directed to a different class of statutory subject matter than claim 1, but recites the same limitations as method claim 1. Since the reference teaches every Limitation of claim 1, it therefore also reads on every limitation of this claim);

a code segment that identifies a context within the input specification (E.g., see col. 6:45, this claim is directed to a different class of statutory subject matter than claim 2, but recites the same limitations as method claim 2. Since the reference teaches every limitation of claim 2, it therefore also reads on every limitation of this claim);

a code segment that parses the identified context to obtain sentences (E.g., see col. 14:24-25, this claim is directed to a different class of statutory subject matter than claim 3, but recites the same limitations as method claim 3. Since the reference teaches every limitation of claim 3, it therefore also reads on every limitation of this claim);

a code segment that determines whether the obtained sentences are testable assertions wherein each testable assertion is a sentence that describes behavior of a program that can be

Art Unit: 2191

tested (E.g., see col. 6:28-30, this claim is directed to a different class of statutory subject matter than claim 1.3, but recites the same limitations as method claim 1.3. Since the reference teaches every limitation of claim 1.3, it therefore also reads on every limitation of this claim);

a code segment that adds (include) the testable assertions to an assertion result set (a list of desired system elements that are included in a test plan), wherein the assertion result set can be used to facilitate (allow) testing (verifying) of the specification (E.g., see col. 2:18-21 , a test index identifying system elements tested by the test code is generated and incorporated into the test plan, allowing the user to verify that all desired system elements are exercised by the automated test code).

Pavela does not disclose application programming interface.

However, MPCD discloses application programming interface (p. 28). Therefore, it would have been obvious to incorporate the teaching of MPCD into Pavela to have the testable assertion describes behavior of application programming interface that can be tested because one would want to test application programming interface as well as other sections of the computer program.

With respect to claim 9, Pavela also discloses a computer program further comprising a code segment that filters (scans) the identified context prior to parsing (interpreting/translating) the context (E.g., see FIG. 3:319,325, col. 6:45-50, tags are scanned to identify system elements, col. 6:24-44, tags are interpreted, translated, and conversational language statement are generated).

With respect to claim 10, Pavela further discloses the code segment that receives the specification is defined to receive the specification in a text format (E.g., see FIG. 6 and 8, col.

Art Unit: 2191

6:6-9 test objects which is operable of performing an assigned function in support of the automated test procedure; col.7:34-35 test code has been generated can be executed).

With respect to claim 14, Pavela discloses automated acquisition of assertions in a specification of a computer program, comprising: a code segment that receives an input specification, wherein the input specification comprises a plurality of sentences (E.g. see col. 2:12-17, this claim is directed to a different class of statutory subject matter than claim 1.1, but recites the same limitations as method claim 1.1. Since the reference teaches every limitation of claim 1.1, it therefore also reads on every limitation of this claim;

a code segment that obtains a sentence from the plurality of sentences (E.g., see col. 6:4, this claim is directed to a different class of statutory subject matter than claim 1.2, but recites the same limitations as method claim 1.2. Since the reference teaches every limitation of claim 1.2, it therefore also reads on every limitation of this claim;

a code segment that determines whether the obtained sentence is a testable assertion, wherein the testable assertion describes behavior of a program that can be tested (E.g., see col. 6:28-30, this claim is directed to a different class of statutory subject matter than claim 8, but recites the same limitations as method claim 8. Since the reference teaches every limitation of claim 8, it therefore also reads on every limitation of this claim;

and a code segment that marks the obtained sentence as testable when the obtained sentence is a testable assertion (E.g., see col. 11:30-32., 2:18-21, this claim is directed to a different class of statutory subject matter than claims 4 and 8, but recites the same limitations as method claims 4 and 8. Since the reference teaches every limitation of claims 4 and 8, it therefore also reads on every limitation of this claim).

Pavela does not disclose application programming interface.

However, MPCD discloses application programming interface (p. 28). Therefore, it would have been obvious to incorporate the teaching of MPCD into Pavela to have the testable assertion describes behavior of application programming interface that can be tested because one would want to test application programming interface as well as other sections of the computer program.

With respect to claim 15, Pavela also discloses a computer program further comprising a code segment that identifies a context within the specification (E.g., see col. 6:45, this claim is directed to a different class of statutory subject matter than claim 2, but recites the same limitations as method claim 2. Since the reference teaches every limitation of claim 2, it therefore also reads on every limitation of this claim).

Claim 11 and 18 are both rejected on the same base of claim 5.

Claim 12 and 19 are both rejected on the same base of claim 6.

Claim 13 and 20 are both rejected on the same base of claim 7.

Claim 15 is rejected on the same base of claim 2.

Claim 16 is rejected on the same base of claim 3

Claim 17 is rejected on the same base of claim 4.

(10) Response to Argument

Appellant has argued, in substance the following:

Art Unit: 2191

(A) Regarding independent claims 1, 8, and 14, as noted on page 13, last paragraph, “Pavela does not teach receiving a specification of a computer program as an input.” “The source file of Pavela is not equivalent to a specification of a computer program.” “The source file of Pavela is not described as including a plurality of sentences describing a computer program.” “It should be further noted that a tag or test code object identifier within the source file of Pavela is not equivalent to a sentence or a context, as required by the present invention.”

On page 14, first full paragraph, Applicant points to Table 1 (found on page 20 of Specification) as an example of the specification of a computer program is a conversant textual description of the computer program’s functionality.

Examiner’s Response:

Examiner respectfully disagrees with Applicant’s assertion that Pavela does not teach receiving a specification of a computer program as an input. Pavela clearly discloses this limitation at col. 2 lines 12-17, the source file includes a member of a library of executable code objects defining a set of instructions (a specification of computer program as an input). It should be noted that the Specification fails to explicitly define “a specification of a computer program.” See Specification, page 6, line 7, “wherein the input specification comprises a plurality of sentences.” See Specification, page 11, line 16, “The specification 102 can be any specification, such as a Javadoc specification of a Java technology API.” Examiner notes that Table 1 is ‘An exemplary context specification’, which is defined (page 19, line 15) as ‘a portion of the specification that is associated with a particular context.’ Table 1 fails to convince Examiner that a specification of a computer program must be a conversant textual description. Pavela’s disclosure appears to be

Art Unit: 2191

analogous. Also note FIG. 6, #602, which shows a code segment for the input specification, further supporting Examiner's position.

(B) Regarding independent claims 1, 8, and 14, as noted on page 15, last paragraph, "Pavela does not teach determining whether a sentence obtained from the specification of the computer program is a testable assertion, wherein the testable assertion describes behavior of an application programming interface that can be tested.

Examiner's Response:

Examiner disagrees. Pavela clearly discloses determining whether a sentence obtaining from the specification of the computer program is a testable assertion, wherein the testable assertion describes behavior of program that can be tested at col. 2 lines 12-17 and col. 6 lines 28-30. MPCD is cited to clarify the term 'application programming interface', defined as "A set of routines used by an application program to direct the performance of procedures by the computer's operating system." Thus, an "application programming interface" is code as disclosed in the Pavela reference. One cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). Tags, as disclosed by Pavela, are associated with a member of a library of executable code objects defining a set of instructions for performing a portion of the automatic test procedure. Thus tags do relate to a testable assertion describing behavior that can be tested.

Art Unit: 2191

(C) Regarding independent claim 1, as noted on page 17, 3rd paragraph, “It is not reasonable to conclude that Pavela teaches marking the obtained sentences as testable when the obtained sentence represents a testable assertion.”

Examiner’s Response:

Examiner disagrees. Pavela discloses FIG. 3:320, col. 6:47 “Tags in the source file 318 are scanned to identify system elements 132 that are tested by the test case, and to generate a test index (the index is automatically generated) from the identified system elements 132.” Thus obtained ‘sentences’ are marked as testable when the obtained sentence represents a testable assertion.

(D) Regarding independent claim 8, as noted on page 17, last paragraph, “Pavela does not teach identification of a context within the specification of the computer program. Pavela does not disclose parsing the context within the specification of the computer program to obtain sentences.”

Examiner’s Response:

Examiner disagrees. A code segment (a scanning operation) identifies a context (tags within the source file) within the input specification (E.g., see col. 6:45). Pavela disclosed (col. 6, lines 45-46), “The tags in the source file 318 are scanned to identify (parsed) system elements 132 (context within the specification of the computer program) that are tested by the test case...”

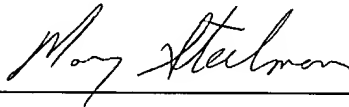
Art Unit: 2191

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

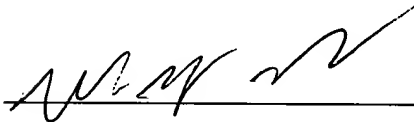
For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,



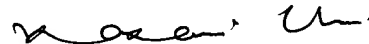
Mary Steelman, Examiner

Conferees:



Wei Zhen, Supervisory Primary Examiner

WEI ZHEN
SUPERVISORY PATENT EXAMINER



Kakali Chaki, Supervisory Primary Examiner

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100